Academic Literacies in Computing at The UWI[1]

DANIEL T. FOKUM, DANIEL N. COORE, AND **GUNJAN MANSINGH**

Abstract

Academic literacy involves writing well, reading comprehension and critical thinking. At The University of the West Indies (UWI), Foundation courses are one vehicle for developing academic literacy. We examined the relationships between the average performance in UWI Foundation courses and performance in four core advanced Computing courses at UWI as well as the overall average Computing performance. Of the four core courses examined, two are heavily quantitative with minimal writing, and the other two involve projects that require substantial reports to be written. Interestingly, there were statistically significant associations between performance in the UWI Foundation courses and the quantitative core Computing courses and also the overall Computing performance, but not with one of the project courses. We present the data supporting these findings and explore the reasons that might explain these observations.

Keywords: Foundation courses; association; Computing courses

Introduction

THE TERM "ACADEMIC LITERACIES" WAS ORIGINALLY COINED to deal with the study of literacies in higher education (Lea and Street 2006). Lea and Street (2006) state that academic literacies deal "with meaning making, identity, power, and authority . . . what counts as knowledge in any particular academic context" (369). At The University of the West Indies (UWI), academic literacies are honed in the context of the Foundation courses. Computing students from the Faculty of Science & Technology are required to take one of FOUN1014 (Critical Reading and Writing in Science & Technology and Medical Sciences) or FOUN1019 (Critical Reading and Writing in the Disciplines). Students are not allowed to graduate from The

1. The views expressed in this paper do not represent the position of the Department of Computing.

UWI without meeting this requirement. The course description for FOUN1014 states that it

> . . . will allow students to engage with the reading and writing practices of, and produce documents relevant to, their disciplines. Students will focus on issues of 'knowing and doing' in science: questions asked, methods of collecting, evaluating and interpreting evidence, and communicating knowledge with accuracy and clarity in writing and oral presentations.

From this description it is clear that this course addresses academic literacy as identified by Lea and Street (2006).

Academic literacy is more than just writing well. It also includes "understanding the task at hand, reading comprehension and critical thinking skills" (Skinner and Mort 2009, 1). Understanding the task at hand and reading comprehension are especially important skills in Computing, as students frequently have to read project specification documents and possibly starter code and then implement some new functionality that is part of the specification.

This paper examines whether a student's performance in certain Computing courses is associated with his or her performance in Foundation courses. A negative result would indicate that there is some sort of misalignment between the Foundation courses and the Computing courses, since the Foundation courses aim to impart academic literacy to their students and the general skills developed by them are also used in the assessments of the Computing courses. Regardless of the order in which students took Foundation and Computing courses, if the same skill sets were being exercised, developed and ultimately assessed, one would expect there to be an association between the performances in these courses. We compared the average performance in Foundation courses with the overall average grade in Computing courses; and we also made the comparison with some individual Computing courses. We selected four Computing courses – three Level 2 courses and one Level 3 course. The three Level 2 courses include one that is mathematics-based, an algorithmic, and one in which students document client requirements in order to build a piece of software. The Level 3 course, on the other hand, has a strong report-writing component. These courses were chosen because they are all core. In addition, two of them require a significant amount of writing, while the rest do not.

The rest of this paper is laid out as follows: a review of related literature; a description of our data sources and how data was prepared for analysis; presentation of the results of our analysis, and finally, some discussion and concluding remarks.

Literature Review

Writing skills are very important to Computing students and graduates. The ACM/IEEE-CS curriculum recommendations have Professional Communication listed as a topic with the associated learning outcome: "Write clear, concise, and accurate technical documents following well-defined standards for format and for including appropriate tables, figures, and references" (ACM/IEEE-CS Joint Task Force on Computing Curricula 2013). In addition, Rauchas *et al.* (2006) found that in South Africa success in high school English was associated with success in tertiary Computer Science. Even though curriculum guidelines, instructors, and educators agree that writing is essential for the domain, there is a lack of emphasis on writing in the computer sciences (Wong 2018). Over the years Computing educators have sought to infuse writing into their courses. In this section we document a number of these efforts.

Some researchers have observed that historically, written communication skills are not emphasised in undergraduate Computer Science (CS) programmes (Hartman 1989; Martin and Winklmann 1995; Börstler and Johansson 1998). In addition, it has been argued that when literacy support is provided outside the curriculum, it has limited success, as it does not have professional context (Skinner and Mort 2009). In order to address writing deficiencies in CS programmes some educators have introduced writing assignments in the contexts of a Data Structures course (Hartman 1989) and one on Artificial Intelligence and Algorithms (Martin and Winklmann 1995). Hartman (1989) observed that these writing assignments gave students the opportunities to improve their writing and to recognise that writing is important to the discipline.

There are different ways of incorporating writing in technical courses. One approach is to have the main course instructors partner with staff from a university's writing centre to teach students how to communicate ideas to both technical and non-technical audiences (Martin and Winklmann 1995; Skinner and Mort 2009). An alternative approach is to have Computing faculty teach the course (Kay 1998; Kaczmarczyk 2003). Yet another approach is to have writing as part of CS courses (Hartman 1989; Walker 1998; Santore and Lorenzen 2009; Grosz et al. 2019). The advantage of this last approach is that the writing has context, and students recognise that writing is important because their instructors, whom they tend to respect technically, are also teaching that good writing is of value.

The case has been made in the literature that when Computer Science students receive graded writing assignments, they recognise that writing is important to the discipline (Hartman 1989; Walker 1998; Kay 1998; Santore and Lorenzen 2009). Written assignments can be included in courses using the "writing across the disciplines" approach, where instructors in all subject areas are encouraged to incorporate writing assignments in their courses (Kay 1998). It is generally desirable that these writing assignments should require revision (Kay 1998; Michael 2000; Kaczmarczyk 2003) and possibly peer editing (Kay 1998). Researchers have shown that when students receive prompt in-depth feedback on their weekly writing assignments, they show more confidence in their writing ability (Minnes et al. 2018).

As already mentioned, frequent graded writing assignments can result in improved student writing quality. For writing assignments to be useful in improving the quality of student writing, students need to get timely feedback on their writing drafts. However, this can be challenging especially when enrolment grows. One of the approaches for managing the grading load is specifications grading. With specifications grading, students are given clear ideas of what constitutes a passing effort for a part of the assignment. If students meet that standard, then they receive a passing grade; otherwise, they do not (Mirsky 2018). Mirsky (2018) describes the use of specifications grading to assess student writing samples in a computer architecture course. She found that using specifications grading resulted in a clear improvement in student writing quality over a semester. In addition, by the end of the course students felt that they were more competent writers.

The literature discussed thus far has shown that writing is important to Computing education, however, in the 80s and 90s many Computing educators realised that written communication skills are not emphasised, and so they began efforts to infuse writing directly into Computing courses. Computing educators have also observed that the writing assignments are most effective when they are developed by them with the support of Writing Centre staff.

In the next section we will introduce the data fields that we retrieved in order to study the effectiveness of The UWI's writing courses for Computing students.

Data Sources and Methodology

The data in this paper come from the student registration system at The UWI, Mona. Queries were written to select all students who had registered for a first-year course in Computing between Semester I 2014/15 and Semester II 2018/19. This query returned over 2200 records, with the following attributes for each row:

- a list of all Computing courses attempted along with matching grades and the years and semesters in which the courses were taken;
- a list of all CXC CAPE (Caribbean Advanced Proficiency Examination) subjects attempted along with matching grades and the year in which the examinations were sat;
- average of all the student's CAPE grades;
- number of credits attempted;
- number of credits passed.

These results were then merged with the Foundation course grades received by these students between Semester I 2014/15 and Semester II 2018/19. For each of the Computing core courses, as well as FOUN1014 (Critical Reading and Writing in Science & Technology and Medical Sciences) and FOUN1101 (Caribbean Civilization), the student's average grade in each course was computed. For each student the overall average Computing grade was computed across all the Computing courses that had been completed by the time the query was run. We similarly generated the overall average Foundation grade. The analysis presented in this paper is based on students who had attempted, but not necessarily passed, at least 15 credits of Computing courses. The threshold of 15 credits was chosen because Computing majors at The UWI, Mona have a total of 15 credits at Level 1. Students who have attempted at least 15 credits are highly likely to be majoring in one Computing discipline. With the data described above we could reason about the trends that existed between performance in Foundation and Computing courses.

Our data were analysed using R (R Core Team 2014) scripts. Our first task was to split the average grade in the Foundation and average GPA in Computing courses into three bins, i.e., F, C/B-, or B/A. These bins correspond to an average GPA for the courses of less than 2, between 2 and 3, and greater than 3. The main test that we used was Pearson's chi-squared test of independence. With this statistical test, the null hypothesis is that the observed grade in the Computing course(s) is independent of the grade achieved in the Foundation course(s). Our significance level for each of the tests was 0.05, meaning that if the observed p-value was less than that, we would reject the null hypothesis and conclude that there was indeed a dependency between the Computing course(s) and the Foundation course(s).

Results

Figures 1 to 5 are histograms showing the distribution of grades for the overall Computing, overall Foundation, and then for the individual Foundation courses. For the histograms we used The UWI's bins from the revised 2014/15 grading policy, with one caveat – the A-, A, and A+ were all lumped into one bin, which we called 'A'. Figure 1 shows that the three most common average Computing grades are F3, C+, and A, with the modal grade being a C+.

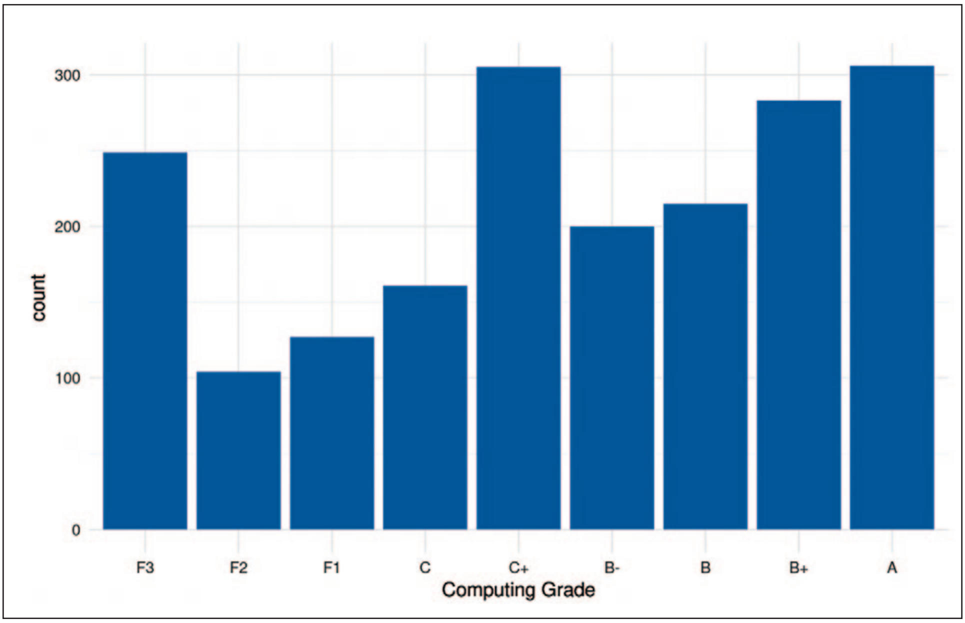

Figure 1. Distribution of average Computing grade per student

Figures 2–5 show that the most common grade in the Foundation courses is a C+. This observation seems to suggest that most students are probably doing just enough to pass the Foundation courses and are not really in pursuit of higher grades. The observation about the modal grade confirms Gosse (2011), which found that the C+ was the most common grade in the Caribbean Civilization course in 2009. It should be noted that the y-axes in Figures 2–5 are plotted on a logarithmic scale given that there are frequently more than three times as many students with C/C+ grades as there are of the next most common grade. A logarithmic axis allows one to visualise the differences better.

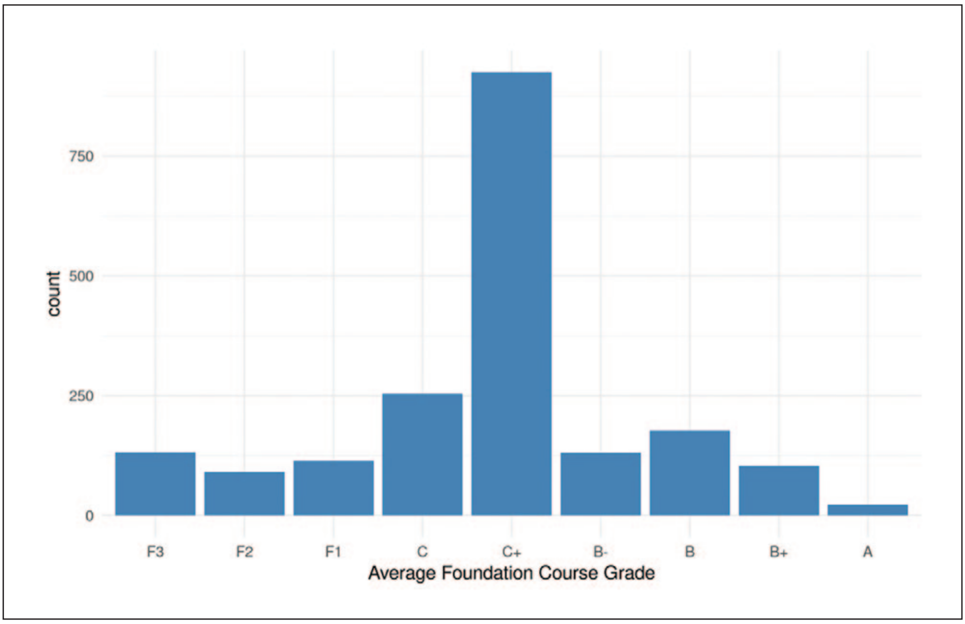

Figure 2. Distribution of average Foundation Course grade per student

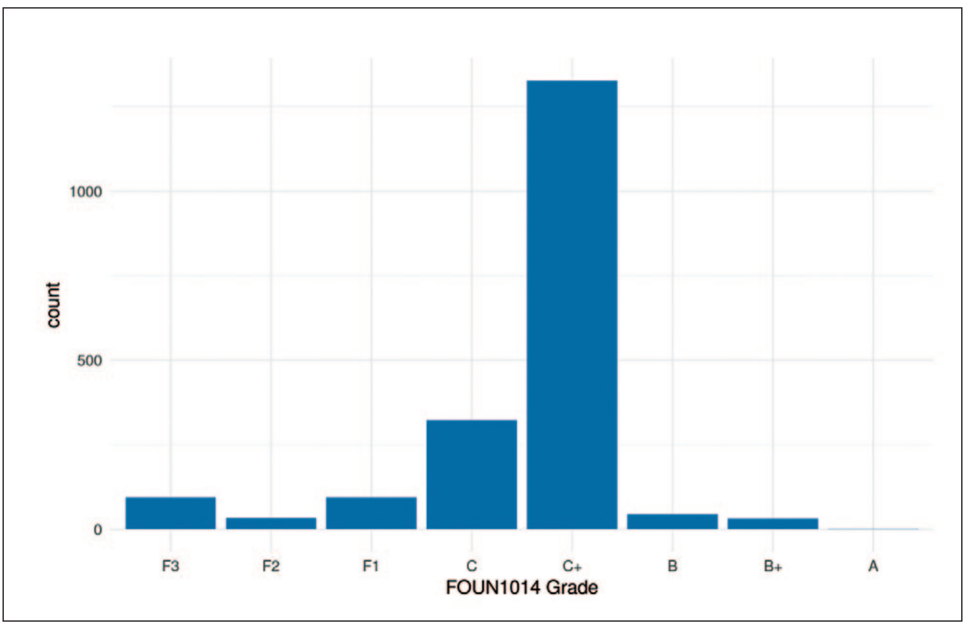

Figure 3. Distribution of average FOUN1014 (Critical Reading and Writing in Science and Technology and Medical Science) grade per student

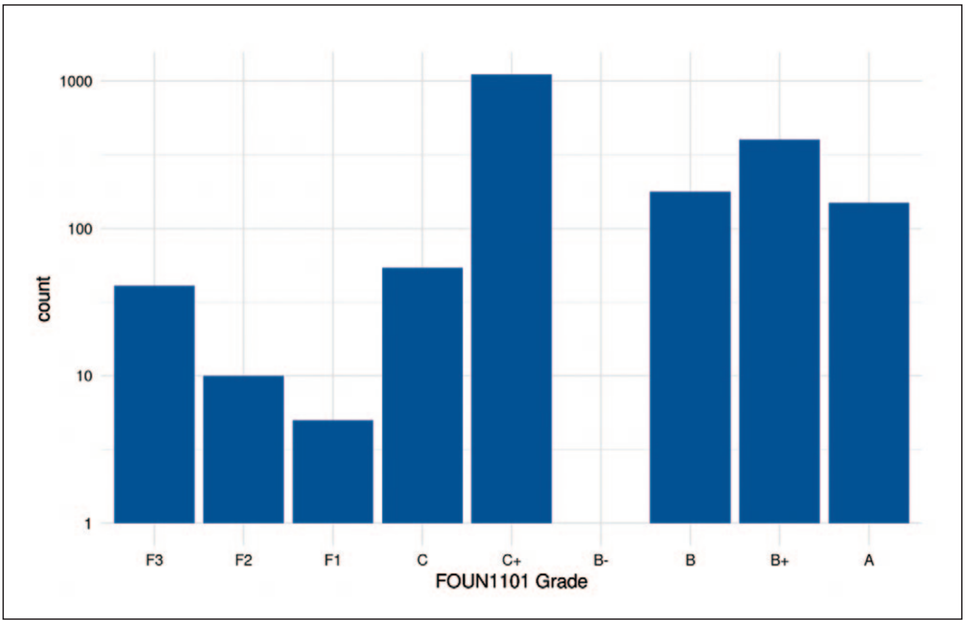

Figure 4. Distribution of average FOUN1101 (Caribbean Civilization) grade per student

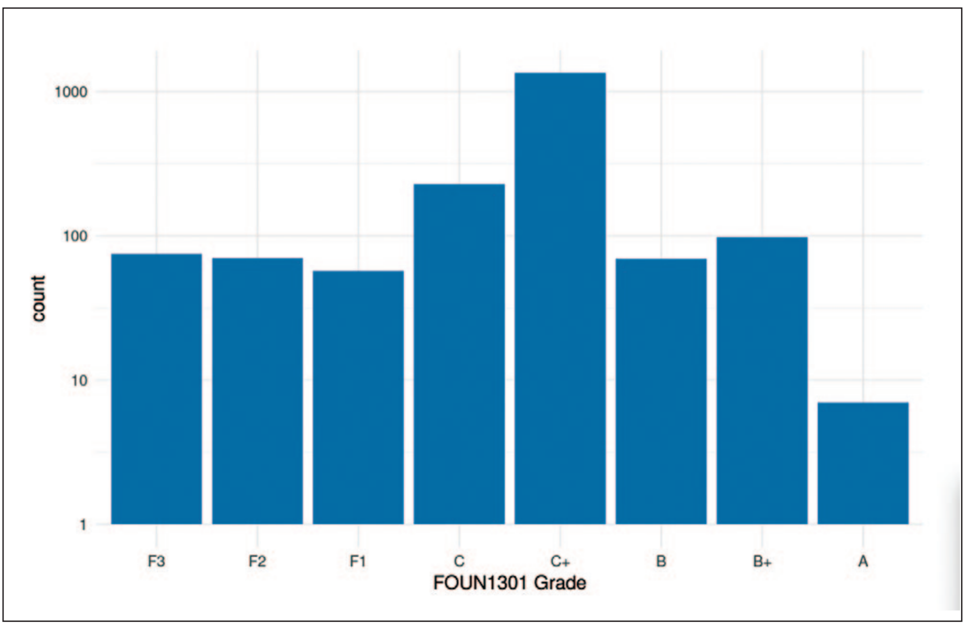

Figure 5. Distribution of average FOUN1301 (Law, Governance, Economy and Society) grade per student

Our next task was to investigate the relationship between performance in Foundation courses, and all Computing courses. Students in the Faculty of Science & Technology are required to take either FOUN1014 (Critical Reading and Writing in Science & Technology and Medical Science) or FOUN1019 (Critical Reading and Writing in the Disciplines) plus at least one of FOUN1301 (Law, Governance, Economy & Society) or FOUN1101 (Caribbean Civilization).

Table 1 is a contingency table summarising the relationship between performance in the Foundation and Computing courses. A chi-squared test for independence returned a simulated p-value <0.000. This result shows that the grades in the Foundation and Computing courses are not independent.

Table 1. Relationship between average grades in all Foundation courses and all Computing courses

	Average Computing Grade			
Avg. Foundation Grade	**F**	**C/B-**	**B/A**	**p-value**
F	43.6%	39.8%	16.6%	< 0.000
C/B-	23.3%	35.0%	41.8%	
B/A	9.2%	24.4%	66.3%	

Table 1 shows that there is statistically significant association between performance in the Foundation courses and all the Computing courses. However, we wanted to drill down further to see if this claim held for all Computing courses. To answer this question, we focused on two writing-intensive Computing courses – COMP2140 (Software Engineering) and COMP3901 (Group Project). In COMP2140, 15% of the final grade is based on the requirements documentation, while another 15% is based on presentations. Thus, 30% of the final grade depends on communication skills.

Table 2 shows that there is statistically significant association between performance in the Foundation courses and COMP2140 performance. A chi-squared test for independence returned a simulated p-value <0.000.

Table 2. Relationship between average grades in all Foundation courses and COMP2140 (Software Engineering)

	Average Computing Grade			
Avg. Foundation Grade	**F**	**C/B-**	**B/A**	**p-value**
F	22.2%	45.6%	32.2%	< 0.000
C/B-	9.7 %	34.8%	55.5%	
B/A	4.3%	29.8%	62.2%	

Table 3. Relationship between average grades in all Foundation courses and COMP3901 (Group Project)

	Average Computing Grade			
Avg. Foundation Grade	F	C/B-	B/A	p-value
F	0.0%	17.2%	82.8%	< 0.000
C/B-	2.3%	23.7%	74.0%	
B/A	0.0%	5.9%	94.1%	

In COMP3901, 50% of the final grade depends on the final report, and 15/50 of the marks for the report are attached to the quality of the language used. It is unusual for students to fail the Group Project course, hence the very low ratios under the F column for the average COMP3901 grade in Table 3 above.

A chi-squared test of independence showed that the average grades for the Foundation courses and the Group Project were statistically independent, with a simulated p-value of 0.108. In the next section we provide some discussion as to why the Foundation courses do not appear to have much effect on the Group Project grade.

We also considered the relationship between the Foundation courses and two technical, but not writing-intensive courses – COMP2201 (Discrete Mathematics for Computer Science) and COMP2211 (Analysis of Algorithms).

Table 4 is a contingency table showing the relationship between the average score in Foundation courses and the average COMP2201 grade.

From Table 4 we see that students whose average Foundation course grade was an F were also more likely to have a failing COMP2201 grade. Students who averaged a B or better in their Foundation courses were also more likely to get a B or better in Discrete Mathematics. Further analysis using a chi-squared test of

Table 4. Relationship between average grades in all Foundation courses and COMP2201 (Discrete Mathematics for Computer Science)

	Average Computing Grade			
Avg. Foundation Grade	F	C/B-	B/A	p-value
F	42.9%	33.3%	23.8%	< 0.000
C/B-	14.1%	39.4%	46.5%	
B/A	9.4%	21.2%	69.4%	

independence returned a simulated p-value <0.000, thus showing that there is statistically significant association between performance in Foundation courses and the Discrete Mathematics course.

Table 5 is a contingency table showing the relationship between the average grade in Foundation courses and the average COMP2211 (Analysis of Algorithms) grade.

Table 5. Relationship between average grades in all Foundation courses and COMP2211 (Analysis of Algorithms)

	Average Computing Grade			
Avg. Foundation Grade	**F**	**C/B-**	**B/A**	**p-value**
F	40.0%	32.7%	27.3%	< 0.000
C/B-	30.5%	33.6%	35.8%	
B/A	15.0%	25.0%	60.0%	

The same trends that we observed with COMP2201 (Discrete Mathematics for Computer Science) also hold true for COMP2211. A chi-squared test of independence returned a simulated p-value of 0.002, thus showing that there is statistically significant association between performance in Foundation courses and the Analysis of Algorithms course.

Discussion and Conclusion

The two results gleaned from Tables 4 and 5 were surprising given that COMP2201 and COMP2211 are not writing-intensive courses. However, the most plausible explanation for this result is that good students tend to do well in many of their courses, while weaker students tend to struggle in many courses.

COMP2140 (Software Engineering) and COMP3901 (Group Project) both have writing components, so it is natural to assume that there would be a strong association between performance in the Foundation courses and in these courses. The data for COMP2140 bore this out to a reasonable extent (e.g. 62.2 per cent of those earning an average Foundation course score of B or higher also got a B or higher in COMP2140, but having a failing Foundation course average was not a strong predictor of any particular grade in COMP2140).

The data for COMP3901 were quite different. Most students happened to obtain a grade of B or higher in COMP3901, perhaps because it is the capstone project course, and students are encouraged to work on something personally meaningful, so they are probably well motivated to perform. Projects are done in groups of up to four students, and students are free to divide the work as they choose. It is quite common for some students to contribute minimally to the development of the software for the project, but to compensate by contributing more substantially to the written report. Regardless of the division of labour, it is typical for multiple members of each group to pay attention to the quality of the report. This means that students who may have performed poorly in their Foundation courses will still likely have a group member who performed well and who contributes to the quality of the report. This might be the reason for the low predictive power of a student's average Foundation course grade and his/her COMP3901 (Group Project) grade.

Interestingly, the data shows that good performance in the Foundation courses is associated with good performance in Computing courses overall. This was borne out directly by the data from Table 1, but more specifically, even in courses that do not have a significant writing component, such as COMP2201 (Discrete Mathematics for Computer Science) and COMP2211 (Analysis of Algorithms), the association with Foundation course performance was statistically significant. COMP2201 is a course on Discrete Mathematics and COMP2211 is a course on the analysis of algorithms; both are deeply technical courses that demand a careful reading of problem information in order to formulate, and then execute, a viable solution. Even though those solutions are not expressed as prose, but rather as mathematical expressions or as computer programs, the same clarity of thought that is required to express those ideas is also required for good writing. Hence the existence of the association is understandable.

We note finally that these associations have no implication for causality, so while we have established associations between average performance in Foundation courses with performance in Computing courses (both with and without significant writing components) this does not indicate that one causes the other. For example, it is quite possible that students who have good acumen and study skills perform well at any subject they apply themselves to, be it a Foundation course with a significant writing component, or a Computing course with a strong demand on logical reasoning. Nevertheless, the data suggest that requiring students to engage in writing exercises does not do harm to their Computing performance. At present, the writing requirement within the Computing curricula is fairly small; only COMP2140 and COMP3901 require significant writing. So, from the point of view of producing well-rounded students who can communicate effectively and demonstrate competence at solving computing problems, it makes sense to develop the writing component within the curricula of the Computing degrees.

References

ACM/IEEE-CS Joint Task Force on Computing Curricula. 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science.* New York: Association for Computing Machinery. https://doi.org/10.1145/2534860

Börstler, Jürgen, and Olof Johansson. 1998. "The Students Conference — a Tool for the Teaching of Research, Writing, and Presentation Skills." In *Proceedings of the 6th Annual Conference on the Teaching of Computing and the 3rd Annual Conference on Integrating Technology into Computer Science Education: Changing the Delivery of Computer Science Education*, 28–31. ITiCSE '98. New York, NY, USA: ACM. https://doi.org/10.1145/282991.283007

Gosse, David. 2011. "The Performance of Students in a Foundation Course in Caribbean Civilization." *The UWI Quality Education Forum* (17): 86–100.

Grosz, Barbara J., David Gray Grant, Kate Vredenburgh, Jeff Behrends, Lily Hu, Alison Simmons, and Jim Waldo. 2019. "Embedded EthiCS: Integrating Ethics Across CS Education." *Communications of the ACM* 62 (8): 54–61. https://doi.org/10.1145/3330794

Hartman, Janet D. 1989. "Writing to Learn and Communicate in a Data Structures Course." *SIGCSE Bulletin* 21 (1): 32–36. https://doi.org/10.1145/65294.71191

Kaczmarczyk, Lisa C. 2003. "A Technical Writing Class for Computer Science Majors: Measuring Student Perceptions of Learning." In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, 341–45. SIGCSE '03. New York: ACM. https://doi.org/10.1145/611892.612003

Kay, David G. 1998. "Computer Scientists Can Teach Writing: An Upper Division Course for Computer Science Majors." *SIGCSE Bulletin* 30 (1): 117–20. https://doi.org/10.1145/274790.273174

Lea, Mary R., and Brian V. Street. 2006. "The 'Academic Literacies' Model: Theory and Applications." *Theory into Practice* 45 (4): 368–77. https://doi.org/10.1207/s15430421tip4504_11

Martin, James H., and Karl Winklmann. 1995. "AIgorithms: An Integrated Algorithm Analysis, Writing and Artificial Intelligence Course." *SIGART Bulletin* 6 (2): 39–41. https://doi.org/10.1145/201977.201995

Michael, Mark. 2000. "Fostering and Assessing Communication Skills in the Computer Science Context." In *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*, 119–23. SIGCSE '00. New York: ACM. https://doi.org/10.1145/330908.331834

Minnes, Mia, Bruce Maxwell, Stephanie R. Taylor, Phillip Barry. 2018. "Writing in CS: Why and How?" In *Proceedings of the 49th SIGCSE Technical Symposium on Computer Science Education*, 402–403. SIGCSE '18. Baltimore: ACM. https://dl.acm.org/doi/pdf/10.1145/3159450.3159620

Mirsky, Grace M. 2018. "Effectiveness of Specifications Grading in Teaching Technical Writing to Computer Science Students." *Journal of Computing Sciences in Colleges* 34 (1): 104–10. http://dl.acm.org/citation.cfm?id=3280489.3280505

R Core Team. 2014. "R: A Language and Environment for Statistical Computing." Vienna, Austria. http://www.R-project.org

Rauchas, Sarah, Benjamin Rosman, George Konidaris, and Ian Sanders. 2006. "Language Performance at High School and Success in First Year Computer Science." In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, 398–402. SIGCSE '06. New York: ACM. https://doi.org/10.1145/1121341.1121467

Santore, John, and Torben Lorenzen. 2009. "Use Writing Class Techniques to Create Software Design Documents." *SIGCSE Bulletin* 41 (2): 136–37. https://doi.org/10.1145/1595453.1595492

Skinner, Iain, and Pam Mort. 2009. "Embedding Academic Literacy Support within the Electrical Engineering Curriculum: A Case Study." *IEEE Transactions on Education* 52 (4): 547–54. https://doi.org/10.1109/TE.2008.930795

Walker, Henry M. 1998. "Writing Within the Computer Science Curriculum." *SIGCSE Bulletin* 30 (2): 24–25. https://doi.org/10.1145/292422.292433

Wong, Melissa. 2018. "We Don't Need to Write to Learn Computer Sciences: Writing Instruction and the Question of First-year, Later or Not-at-all." *Journal of Pedagogic Development* 8 (3): 1–21. https://uobrep.openrepository.com/uobrep/bitstream/10547/622922/1/470-1212-1-SM.pdf